

Мельничук А. С.,  
Луценко С. П.,  
Громовий Д. С.,  
Трофимова К. В.

## АНАЛІЗ МЕТОДІВ СОРТУВАННЯ МАСИВУ ЧИСЕЛ

У даній статті розглянуті методи сортування (розстановки за певним правилом) в масиві чисел, які використовуються в обчислювальній техніці на сьогоднішній день. Існуючі методи аналізуються за показниками кількості обмінів, ітерацій та порівнянь, що найповніше характеризують кожен алгоритм. В результаті оцінена загальна швидкодія кожного методу, визначені переваги та недоліки кожного з них.

**Ключові слова:** сортування, масив чисел, швидкодія, ітерація, об'єм пам'яті, алгоритм, програмування

### 1. Вступ

Алгоритми сортування масиву чисел — одні з найбільш важливих процесів, які використовуються у обчислювальній техніці, оскільки сортування та пошук є найбільш загальними складовими частинами систем програмування [1]. Останнім часом інтерес до методів та засобів сортування великих масивів інформації зростає, що тісно пов'язано з новими досягненнями в області комп'ютерних технологій, а також з розширенням сфери застосування самих алгоритмів сортування [2, 3].

Сортування — це процес розстановки елементів в деякому визначеному порядку, за попередньо заданим законом. При цьому елементи розміщуються таким чином, щоб, по-перше, обчислення, які вимагають певного порядку розміщення даних, могли виконуватися ефективно, по-друге, результати мали змістовний вигляд, і по-третє, наступні процеси мали би придатні вихідні дані [1].

**Мета даної статті** — проаналізувати часові характеристики процесу сортування для вибору алгоритму, найбільш придатного для реалізації конкретної задачі.

### 2. Аналіз літературних даних і постановка проблеми

Незважаючи на значну кількість алгоритмів та вдосконалень відомих методів сортування, основними, або базовими, серед них є сім методів: метод лінійного вибору, лінійного вибору з обміном, лінійного вибору з підрахунком, парного обміну, стандартного обміну, просіювання та лінійної вставки [3].

Існуючі методи сортування, кожен з яких має свої переваги та недоліки [4], можна згрупувати у відповідності з визначальними принципами функціонування у три основні категорії [5]:

- сортування за допомогою включення (by insertion);
- сортування за допомогою виділення (by selection);
- сортування за допомогою обмінів (by exchange).

Двома найбільш важливими програмними аспектами є об'єм машинного часу, необхідний для роботи програми сортування, і об'єм пам'яті, необхідний для цієї програми [1]. Доброю мірою ефективності може бути число необхідних порівнянь та число переміщень (перестановок) елементів масиву. Ці величини є функціями від числа елементів  $n$ , які сортуються.

Ефективні алгоритми сортування потребуються порядку  $n \log_2 n$  порівнянь. В простих і очевидних методах, які називають прямими, потрібно порядку  $n^2$  порівнянь [5].

Для розгляду алгоритмів сортування доцільно ввести деяке умовне позначення [5]. Якщо  $a$  — елементи

$$a_1, a_2, \dots, a_n,$$

то сортуванням є перестановка цих елементів до одержання масиву

$$a_1, a_2, \dots, a_n,$$

в якому при деякій функції впорядкування  $f$  виконується відношення

$$f(a_{k1}) \leq f(a_{k2}) \leq \dots \leq f(a_{kn}).$$

Звичайно, функція впорядкування не обраховується за якимось правилом, а зберігається як дійсна компонента (поле) кожного елемента [5]. Методи сортування можна назвати стійким, якщо в процесі сортування відносне розташування елементів з рівними ключами не змінюється [6].

Проведемо порівняльний аналіз найбільш відомих методів сортування.

### 3. Порівняльний аналіз методів сортування масиву чисел

Найпростіший метод сортування називається бульбашковим сортуванням: на першому проході сусідні елементи порівнюються і якщо елементи не впорядковані, вони змінюються місцями, в результаті чого найбільший із тих, що залишились (порівняння іде між  $(n - 1)$  елементами), переміщується на передостаннє місце і так до тих пір, поки масив не виявиться впорядкованим [5]. Основна перевага бульбашкового методу сортування — це те, що його легко реалізувати у вигляді програми. Недолік цього методу полягає в тому, що він володіє незначною швидкістю [1].

Найбільш істотного покращення можна досягти, використовуючи сортування Шела (або сортування з кроком, що зменшується). В цьому методі в першопочатковому масиві сортуються окремі підмасиви. При цьому  $K$  називається кроком  $n$ , наприклад, якщо  $K$  рівне 5, то першим сортується підмасив, який складається з елементів

$a(1)$ ,  $a(6)$ ,  $a(11)$  і т. д. П'ять підмасивів, кожен з яких містить  $1/5$  елементів початкового масиву, сортуються аналогічним чином [4].

При швидкому сортуванні за методом Хоара масив переглядається одразу з двох боків. Якщо з одного боку знаходиться елемент більший, ніж з іншого, то ці елементи змінюються місцями. Так продовжується до того часу, поки вказівники на лівий і правий елементи не перетнуться. В результаті такого проходження алгоритму весь масив розбитий на дві частини. В лівій частині знаходяться елементи з меншими значеннями ключів, а в правій — з більшими. Тепер залишається відсортувати тим самим методом кожен частину окремо [3].

При сортуванні методом підрахунку кожен елемент порівнюється з усіма іншими елементами. Якщо який-небудь елемент більший, ніж  $K$  елементів цього масиву, то після впорядкування він повинен займати  $(K + 1)$  місце, відповідно, необхідно порівняти попарно всі ключі та підрахувати, скільки з них менше кожного окремого ключа. Після цього необхідно переставити елементи у відповідності з певними місцями [6].

При сортуванні вставкою елементи переглядаються по одному і кожен новий елемент вставляється в підходяще місце серед раніше впорядкованих елементів [3]. Суть сортування простими вставками полягає в наступному: нехай на  $i$ -му кроці алгоритму послідовність з  $(i - 1)$  перших елементів масиву впорядкована. Додавання  $i$ -го елемента в цю послідовність відбувається впорядковано: елемент, який потрібно вставити, порівнюється з попереднім до тих пір, поки не знайдеться місце  $i$ -го елемента [5].

Сортування простим вибором здійснюється так: спочатку з  $n$  елементів вибирається максимальний. Далі  $n$ -ний та знайдений максимальний елементи міняються місцями. Потім серед  $(n - 1)$  елементів, що залишилися, від 1 до  $(n - 1)$  вибирається максимальний і міняються місцями з елементом, який стоїть на  $(n - 1)$ -му місці і т. д. [7]. Так продовжується до того часу, поки весь масив не буде відсортований. Останній раз максимум вибирається з двох елементів, які стоять на першому і на другому місці відповідно, і більший з них стає на друге місце [5].

Сортування за допомогою двійкового дерева пошуку, яке можна використовувати для сортування, виконується так: береться порожнє дерево, до нього додаються всі елементи масиву в порядку зростання [1]. Якщо елементи масиву різні та розміщені у випадковому порядку, а довжина масиву  $n$ , то алгоритм вимагає в середньому  $n \log_2 n$  операцій [8]. Якщо елементи відсортовані у порядку збільшення або порядку спадання, то дерево стає незбалансованим (тобто у нього з'являється багато порожніх гілок), і алгоритм вимагає  $n^2$  операцій [1].

Алгоритм сортування злиттям є процесом об'єднання двох або більш відсортованих файлів в деякий третій відсортований файл [9]. Нехай є два відсортованих масиви  $M$  і  $N$ , тоді порівнюються два найменші елементи обох масивів і найменший з них виводиться як найменший елемент сумарного масиву, потім процедура повторюється до тих пір, поки результуючий масив не буде відсортований злиттям.

Порівняння здійснювалось за такими характеристиками: кількість ітерацій (КІ), кількість порівнянь в ітерації (КПІ), загальна кількість порівнянь (ЗКП). Для кожного методу наведемо також переваги та недоліки.

#### 1. Метод бульбашки (bubble) [1, 5].

$KI = (n - 1)$ ,  $KPI = (n - 1)$ ,  $ЗКІ = (n^2 - n)/2$ .

*Переваги:* для поліпшеного сортування середнє число проходів (ітерацій) рівне  $n - k\sqrt{n}$ .

*Недоліки:* середня кількість переміщень  $3n(n - 1)/2$ ; велика кількість обмінів при масиві зі зворотнім порядком [6].

#### 2. Метод Шела (Shell) [1, 6].

$KI = \log_2 n$ ,  $ЗКІ = n(\log_2 n)^2$ .

*Переваги:* мінімальний об'єм пам'яті, скорочення повторних порівнянь.

*Недоліки:* велике число обмінів при масиві зі зворотнім порядком.

#### 3. Метод лінійного вибору з підрахунком (linear choice with count) [3, 6].

$KPI = n - 1$ ,  $ЗКІ = n^2/2$ .

*Переваги:* можливість обрахувати адресу будь-якого елемента в списку виведення.

*Недоліки:* необхідна додаткова пам'ять під список лічильників.

#### 4. Метод лінійної вставки (linear insert) [2, 3, 5].

$KPI = n/2$ ,  $ЗКІ = n^2/4$ ,  $ЗКІ = (n^2 - n - 2)/4$  [5].

*Переваги:* всі елементи повинні бути відомі.

*Недоліки:* виконується переміщення з однієї позиції в іншу, а не взаємний обмін; середня кількість переміщень  $(n^2 + 9n - 10)/4$ .

#### 5. Метод просіювання (лінійна вставка з обміном) (sifting) [3, 6].

$ЗКІ = (n - 1)^2/2$ ,  $ЗКІ = n^2/4$  [6].

*Переваги:* метод не зберігає фіксовану послідовність порівнянь.

*Недоліки:* велике число обмінів при списку в зворотному порядку.

#### 6. Метод швидкого сортування, або метод Хоара (quick sort) [4, 5].

В кращому випадку  $ЗКІ = n \log_2 n$ , в гіршому —  $ЗКІ = n^2$  [5, 7].

*Переваги:* в метод можна легко включити будь-який метод сортування.

*Недоліки:* недостатньо висока продуктивність при невеликих  $n$ .

#### 7. Метод простого вибору (simple choice) [1, 5, 9].

$KI = n - 1$ ,  $ЗКІ = n^2$ .

*Переваги:* метод не вимагає використання додаткової пам'яті.

*Недоліки:* середня кількість переміщень  $n(\ln n + g)$ , де  $g = 0.577216$  — постійна Ейлера. Час сортування збільшується квадратично відносно кількості елементів. Метод характеризується нестійкістю [9].

#### 8. Метод підрахунку (calculation) [5, 8].

$ЗКІ = (n^2 - n)/2$ .

*Недоліки:* кількість переміщень  $9n/4$ . Вимагає додаткової пам'яті.

#### 9. Метод бінарного дерева (binary tree) [1, 3, 8].

$ЗКІ = n^2$ ,  $ЗКІ = n \log_2 n$  [8].

*Переваги:* мінімальна кількість рівнів та мінімальний об'єм.

*Недоліки:* вимагає додаткової пам'яті; немає можливості точно передбачати потрібний об'єм пам'яті.

#### 10. Метод простого злиття (simple merge) [1, 8, 10].

$KI = \log_2 n$ ,  $ЗКІ = n \log_2 n$ .

*Недоліки:* потребує додаткової пам'яті; доступ до даних відбувається набагато повільніше, ніж при використанні інших методів.

**11. Метод сортування з обчисленням адреси (sorting with calculation of address) [1, 8].**

$$ЗКІ = n^2.$$

*Переваги:* велика швидкодія.

*Недоліки:* порядок елементів в масиві не збігається з їх порядком в списку; складність здійснення динамічне розширення масиву.

**12. Метод пірамідального сортування (heap sort) [7].**

$$ЗКІ = n \log_2 n.$$

*Переваги:* не потребує додаткової пам'яті.

*Недоліки:* метод характеризується нестійкістю.

**13. Метод порозрядного сортування (radix sort) [1, 10].**

$$ЗКІ = n \log_2 n.$$

*Переваги:* найбільш підходить для використання в комп'ютерній графіці; списки легко реорганізовувати, об'єднувати.

*Недоліки:* вимагає додаткову пам'ять [8].

**14. Метод парного обміну (pair exchange) [3, 6].**

$KI = n$ ; в кращому випадку  $ЗКІ = n$ , в гіршому —  $ЗКІ = n^2/2$ .

*Переваги:* найвища ефективність при використанні у вигляді мережі для паралельних процесорів.

*Недоліки:* потребує додаткових витрат на управління парними-непарними переглядами; великий об'єм пам'яті під процедури і низька швидкодія.

**15. Метод турнірного сортування (tournament sorting) [1, 9].**

$$KI = n - 1, KPI = n - 1, ЗКІ = n \log_2 n.$$

*Переваги:* мінімальний метод по пам'яті.

*Недоліки:* верхні рівні дерева містять вказівники, а дійсні дані зберігаються тільки на найнижчому рівні; на декількох рівнях дерева присутнє дублювання інформації; вимагає додаткового об'єму пам'яті.

Порівняння методів сортування в загальному вигляді представлено в табл. 1, де приведено мінімальне, середнє та максимальне значення вказаних величин [4, 6–10].

Нами було проведено перевірку швидкодії комп'ютерних програм на основі відомих методів сортування.

Час роботи різних програм сортування для масиву розмірністю  $n = 256$  представлено в табл. 2.

**4. Висновки**

Результати аналізу відомих методів сортування за такими часовими характеристиками, як кількість порівнянь в ітерації, загальна кількість порівнянь, кількість ітерацій (переглядів), виконаних в даній роботі, дозволяють вибрати оптимальний з цієї точки зору метод сортування для програмної та апаратної реалізації.

Аналіз сучасних співвідношень при програмній реалізації трьох найбільш поширених груп методів сортування (методи включення, вибору й обміну) свідчить про можливість нових рішень в області апаратної реалізації з метою підвищення рівня паралелізму та пришвидшення процесу сортування. У цьому випадку перевагу слід надати методам сортування на основі обміну, наприклад, методу парного обміну, оскільки він вимагає мінімальних часових витрат (найменшу кількість обмінів). При цьому підтверджено, що саме метод парного обміну можна реалізувати у вигляді сортувальної мережі.

В результаті перевірки відомих методів сортування на швидкодію з використанням масиву сталого об'єму було визначено час роботи програм сортування на основі різних методів. При цьому найкращі результати показали методи Шела та швидкого сортування.

**Література**

1. Лэнгсам, Й. Структуры данных для персональных ЭВМ [Текст] / Й. Лэнгсам, М. Огенстайн, А. Тененбаум. — М. : Мир, 1989. — 568 с.
2. Вышинский, В. А. Сортировка чисел в матрично-алгебраической ЭВМ [Текст] / В. А. Вышинский // Управляющие системы и машины. — 2001. — № 2. — С. 50–52.
3. Лорин, Г. Сортировка и системы сортировки [Текст] / Г. Лорин. — М. : Мир, 1983. — 384 с.
4. Вирт, Н. Алгоритмы + структуры данных = программа [Текст] / Н. Вирт. — М. : Мир, 1985. — 406 с.

**Таблиця 1**

Порівняння методів сортування

Група методів сортування	Мінімальне значення		Середнє значення		Максимальне значення	
	Число необхідних порівнянь ключів	Число обмінів	Число необхідних порівнянь ключів	Число обмінів	Число необхідних порівнянь ключів	Число обмінів
Включення	$n - 1$	$2(n - 1)$	$(n^2 + n - 2)/4$	$(n^2 - 9n - 10)/4$	$(n^2 + n - 2)/2 - 1$	$(n^2 - 3n - 4)/2$
Вибір	$(n^2 - n)/2$	$3(n - 1)$	$(n^2 - n)/2$	$n(\ln n + 0,57)$	$(n^2 - n)/2$	$n^2/4 + 3(n - 1)$
Обмін	$(n^2 - n)/2$	0	$(n^2 - n)/2$	$(n^2 - n)/3/4$	$(n^2 - n)/2$	$(n^2 - n)/3/2$

**Таблиця 2**

Час роботи сортування (в секундах)

Методи сортування	Природа масиву чисел		
	Впорядкований	Випадковий	Зворотний порядок
Лінійна вставка	0,82	0,82	1,64
Лінійний вибір	0,94	0,96	1,18
Бульбашка	1,26	2,04	2,80
Метод Шела	0,10	0,24	0,24
Пірамідальне сортування	0,20	0,20	0,20
Швидке сортування	0,08	0,12	0,08
Злиття	0,18	0,18	0,18

5. Программирование алгоритмов обработки данных [Текст] / О. В. Ускова, Н. В. Огаркова, И. Е. Воронина и др. — СПб. : БХВ-Петербург, 2003. — 102 с.
6. Гузик, В. Ф. Организация различных методов сортировки в вычислительных системах [Текст] / В. Ф. Гузик, В. Е. Золотовский, С. А. Чиненков // Электронное моделирование. — 1992. — Т. 14, № 3. — С. 25–28.
7. Hea, M. An optimal and processor efficient parallel sorting algorithm on a linear array with a reconfigurable pipelined bus system [Text] / M. Hea, X. Wua, S. Q. Zhengb // Computers & Electrical Engineering. — 2009. — Vol. 35, Issue 6. — pp. 951–965.
8. Кнут, Д. Э. Искусство программирования. Т. 3. Сортировка и поиск [Текст] / Д. Э. Кнут. — 2-е изд. — М. : Вильямс, 2003. — 832 с.
9. Chandra, S. Sorting algorithms on transputer arrays [Text] / S. Chandra, M. Jain, A. Basu, P. S. Kumar // Parallel Computing. — 1993. — Vol. 19, Issue 6. — pp. 595–607.
10. Lin, Y.-C. Parallel sorting with cooperating heaps in a linear array of processors [Text] / Yen-Chun Lin, Ferng-Ching Lin // Parallel Computing. — 1990. — Vol. 16, Issue 2–3. — pp. 273–278.

#### АНАЛИЗ МЕТОДОВ СОРТИРОВКИ МАССИВА ЧИСЕЛ

В данной статье рассмотрены методы сортировки (расстановки по определенному правилу) в массиве чисел, которые используются в вычислительной технике на сегодняшний день. Существующие методы анализируются по показателям количества обменов, итераций и сравнений, что наиболее полно характеризуют каждый алгоритм. В результате оценено общее

быстродействие каждого метода, определены преимущества и недостатки каждого из них.

**Ключевые слова:** сортировка, массив чисел, быстродействие, итерация, объем памяти, алгоритм, программирование.

*Мельничук Андрій Сергійович, кафедра радіотехніки, Вінницький національний технічний університет, Україна, e-mail: a\_melnychuk@i.ua.*

*Луценко Сергій Петрович, кафедра радіотехніки, Вінницький національний технічний університет, Україна.*

*Громовий Дмитро Сергійович, кафедра радіотехніки, Вінницький національний технічний університет, Україна.*

*Трофимова Карина Вікторівна, кафедра проектування медико-біологічної апаратури, Вінницький національний технічний університет, Україна.*

*Мельничук Андрей Сергеевич, кафедра радиотехники, Винницкий национальный технический университет, Украина.*

*Луценко Сергей Петрович, кафедра радиотехники, Винницкий национальный технический университет, Украина.*

*Громовой Дмитрий Сергеевич, кафедра радиотехники, Винницкий национальный технический университет, Украина.*

*Трофимова Карина Викторовна, кафедра проектирования медико-биологической аппаратуры, Винницкий национальный технический университет, Украина.*

*Melnychuk Andriy, Vinnytsia National Technical University, Ukraine, e-mail: a\_melnychuk@i.ua.*

*Lutsenko Serhiy, Vinnytsia National Technical University, Ukraine.*

*Hromovyi Dmytro, Vinnytsia National Technical University, Ukraine.*

*Trofymova Karyna, Vinnytsia National Technical University, Ukraine.*

УДК 656.11

**Санько Я. В.,  
Ройко Ю. Я.**

## ДОСЛІДЖЕННЯ ЗАКОНОМІРНОСТЕЙ ФОРМУВАННЯ СТРУКТУРИ ВУЛИЧНО- ДОРОЖНЬОЇ МЕРЕЖІ

В даній статті було визначено межі варіювання оптимальної довжини ділянки вулично-дорожньої мережі в залежності від рівня автомобілізації та кількості мешканців кварталу. Такі результати дають змогу визначити оптимальну довжину ділянки вулично-дорожньої мережі при проектуванні нових сільбищних територій, або навпаки при існуючій транспортній мережі визначити параметри транспортних потоків та житлової забудови.

**Ключові слова:** вулично-дорожня мережа, житловий квартал, транспортний потік, довжина ділянки

### 1. Вступ

Сучасні вимоги щодо переміщення транспортних та пішохідних потоків визначають головні напрямки створення або розвитку раціональної структури вулично-дорожньої мережі, яка б забезпечувала мінімальні витрати коштів на ці переміщення.

Відомо, що створення раціональної структури вулично-дорожньої мережі можна досягти шляхом розробки комплексу транспортних, будівельних, планувальних та природоохоронних заходів. Це дозволить найкращим шляхом вирішити проблему транспортного обслуговування населення міста [1–6].

### 2. Актуальність теми

З історичним розвитком міст, здійснювалося його функціональне призначення, конфігурація вулично-дорожньої мережі, що в свою чергу впливало на геометризовану схему планування міста [7].

Найбільш доцільними в історії проектування та розвитку міст стали квадратно-решітчасті форми міського плану, де основною перевагою є рівномірне дублювання всіх напрямків дорожнього руху без перевантаження міського центру [8].

Із досліджень розподілу сільбищної території та типових геометризованих схем, випливає, що головний акцент приділяється формуванню розгалуженої вулично-дорожньої мережі міста [8–10].